
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Ferranti, Luca; Åström, Kalle; Oskarsson, Magnus; Boutellier, Jani; Kannala, Juho
Homotopy Continuation for Sensor Networks Self-Calibration

Published in:
Proceedings of 29th European Signal Processing Conference, EUSIPCO

DOI:
[10.23919/EUSIPCO54536.2021.9616184](https://doi.org/10.23919/EUSIPCO54536.2021.9616184)

Published: 08/12/2021

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Ferranti, L., Åström, K., Oskarsson, M., Boutellier, J., & Kannala, J. (2021). Homotopy Continuation for Sensor Networks Self-Calibration. In *Proceedings of 29th European Signal Processing Conference, EUSIPCO* (pp. 1725-1729). Article 9616184 (European Signal Processing Conference). IEEE.
<https://doi.org/10.23919/EUSIPCO54536.2021.9616184>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Homotopy Continuation for Sensor Networks Self-Calibration

Luca Ferranti
University of Vaasa
Vaasa, Finland

Kalle Åström
Lund University
Lund, Sweden

Magnus Oskarsson
Lund University
Lund, Sweden

Jani Boutellier
University of Vaasa
Vaasa, Finland

Juho Kannala
Aalto University
Espoo, Finland

Abstract—Given a sensor network, TDOA self-calibration aims at simultaneously estimating the positions of receivers and transmitters, and transmitters time offsets. This can be formulated as a system of polynomial equations. Due to the elevated number of unknowns and the nonlinearity of the problem, obtaining an accurate solution efficiently is nontrivial. Previous work has shown that iterative algorithms are sensitive to initialization and little noise can lead to failure in convergence. Hence, research has focused on algebraic techniques. Stable and efficient algebraic solvers have been proposed for some network configurations, but they do not work for smaller networks. In this paper, we use homotopy continuation to solve four previously unsolved configurations in 2D TDOA self-calibration, including a minimal one. As a theoretical contribution, we investigate the number of solutions of the new minimal configuration, showing this is much lower than previous estimates. As a more practical contribution, we also present new subminimal solvers, which can be used to achieve unique accurate solutions in previously unsolvable configurations. We demonstrate our solvers are stable both with clean and noisy data, even without nonlinear refinement afterwards. Moreover, we demonstrate the suitability of homotopy continuation for sensor network calibration problems, opening prospects to new applications.

Index Terms—homotopy continuation, minimal problems, TDOA, sensor networks calibration

I. INTRODUCTION

Given a network of receivers and transmitters, network self-calibration refers to the simultaneous estimation of positions of both receivers and transmitters [1], [2]. Network self-calibration is essential in several applications, such as beamforming [3] or structure from sound [4]. The case of synchronized networks, where all time delays are known and only the positions of the nodes need to be estimated, has been greatly investigated in the literature [5], [6], [6], [7] and it is mainly solved. The case of *unsynchronized* networks, where also some time offsets of the nodes need to be estimated, has proven more challenging and is still an active research area [8], [9]. In this paper, we focus on 2D self-calibration of networks with synchronized receivers but unsynchronized transmitters, known as network 2D Time Difference Of Arrival (TDOA) self-calibration [10], [11].

Let a network with m receivers and n transmitters, from now on shortly denoted as mr/ns , the TDOA self-calibration can be formalized by a set of mn equations in the form

$$\|\mathbf{r}_i - \mathbf{s}_j\|^2 = (f_{ij} - o_j)^2 = d_{ij}^2, \quad i = 1, \dots, m \quad j = 1, \dots, n \quad (1)$$

where $\mathbf{r}_i, \mathbf{s}_j$ are the unknown positions of receiver i and transmitter j , d_{ij} is the distance between them, o_j is the unknown transmitter offset and f_{ij} is the *pseudorange*, i.e. what the receiver i measures from the transmitter j . To study the solvability of this problem, the *excess constraint* is defined as the number of constraints minus the number of unknowns. For 2D TDOA, the excess constraint is

$$mn - 2m - 3n + 3, \quad (2)$$

where the term $+3$, called *Gauge freedom*, is introduced because only distances are measured and hence the positions can be recovered only up to Euclidean transformation. If the expression in (2) is negative, then we have less constraints than unknowns, and the problem will have infinitely many solutions, if it is zero, then the problem is determined and it will have a finite, but not necessarily unique, number of finite solutions. These determined problems are referred to as *minimal*, because they are the smallest problems to have a finite solution. Finally, if the expression in (2) is positive, then the problem is overdetermined, or *subminimal*, and it will have a unique solution (given zero noise there will be a unique solution that satisfies the constraints exactly, otherwise we find a solution that minimizes the constraint equations).

Given the strong nonlinearity and the high number of unknowns, numerical iterative approaches are very sensitive to initialization and hence perform poorly with a random or grid-searched initial value [12], [13]. Algebraic techniques have proved more efficient for both synchronized [6], [14] and unsynchronized [10], [11], [15], [16] network calibration. These algebraic techniques, based on Gröbner bases, are split into two phases [17]: an offline phase, during which an optimized solver for a specific kind of problem is generated; and an online phase, during which the generated solver can be applied to quickly solve any instance of that problem. While these techniques have proved successful in several applications [18], [19], they are somehow bounded by the complexity of the problem and they have not been able to solve more challenging TDOA configurations. *Homotopy continuation*, a numerical algorithm for systems of polynomial equations [20], has on the other hand been shown to solve more complex problems, where Gröbner bases methods failed [21], [22]. In this paper, we demonstrate that homotopy continuation can also be applied to self-calibration of sensor networks, presenting stable solvers for several previously unsolvable

TABLE I: Different TDOA configurations for m receivers and n transmitters. X: previously solved. O: solved in this paper. *: reducible to a solved configuration. -: underdetermined. u: unsolved. Underlined the minimal configurations

		n			
		3	4	5	6
m	4	-	-	<u>u</u>	u
	5	-	O	<u>O</u>	X
	6	<u>O</u>	O/X	*	*
	7	<u>O</u>	X	*	*
	8	*	*	*	*
	9	X	*	*	*

configurations. To the best of our knowledge, we are the first to apply homotopy continuation to sensor networks self-calibration.

The contribution of our paper is mainly two-fold. First, we show that homotopy continuation can robustly solve sensor network localization problems, opening frontiers for even more applications in this domain. Secondly, through homotopy continuation we present several new algebraic solvers for network configurations that were previously unsolvable. Particularly, we present solvers for the following previously unsolvable configurations (also summarized in Table I).

- **6r/3s**: this is a minimal configuration, i.e. it is the smallest network to have a finite number of solutions. The solution, however, is not unique. From a theoretical perspective, it is interesting to investigate how many solutions the problem can have. It was speculated in [23], that this configuration could have up to 150 distinct solutions, but some of those might be false roots. Using our homotopy solver, we are able to give empirical evidence that the number of solutions is actually much smaller.
- **6r/4s** and **7r/3s**: these can be regarded as the subminimal configurations of the previous ones, obtained adding one extra point to achieve uniqueness of the solution. While the 7r/3s solver is new, an algebraic solver for 6r/4s was already presented in [16]. This, however, required an extra nonlinear refinement of the solution, while our homotopy continuation solver is already stable without this post-processing step.
- **5r/4s**: This configuration has an interesting property, while it is not minimal, it is the only subminimal configuration which cannot be reduced to a minimal one. Hence, it requires its own specialized solver.
- **5r/5s**: This solver can be considered the subminimal configuration of 4r/5s (which remains unsolved), with an extra point to ensure the uniqueness of the solution. Even if a solver for the 4r/5s were found, it would still have multiple solutions. This solver, hence, is the smallest solver to achieve a unique solution in a network with five transmitters.

This paper is structured as follows: The theoretical background of homotopy continuation is discussed in Section II and the algorithms for network calibration are presented in

Section III. Finally, the results are discussed in Section IV and conclusions are drawn in Section V.

II. BACKGROUND: HOMOTOPY CONTINUATION

Homotopy continuation is an iterative algorithm to solve systems of polynomial equations [20] and it has several success stories in e.g. computer vision [21], [24], signal processing [25], [26], and process design [27], [28]

Let $F(\mathbf{x}) = [f_1, \dots, f_n]^T$ and $G(\mathbf{x}) = [g_1, \dots, g_n]^T$ be vectors of polynomials from the ring $\mathbb{C}[x_1, \dots, x_n]$. Suppose we want to solve the system $F(\mathbf{x}) = \mathbf{0}$ and that the system $G(\mathbf{x}) = \mathbf{0}$, referred to as *starting system*, can be solved easily and has at least as many distinct roots as F . We can now define the *homotopy*

$$H(\mathbf{x}, t) = (1 - t)F(\mathbf{x}) + \gamma tG(\mathbf{x}), \quad (3)$$

where t is a new variable and γ is a complex number with $\|\gamma\| = 1$, introduced for numerical stability reasons [20]. It is now easy to see that $H(\mathbf{x}, 1) = \gamma G(\mathbf{x})$ and $H(\mathbf{x}, 0) = F(\mathbf{x})$. Also if \mathbf{x}_0 is a root of G , then $H(\mathbf{x}_0, 1) = \mathbf{0}$ and if \mathbf{x}^* is a root of F , then $H(\mathbf{x}^*, 0) = \mathbf{0}$.

The core idea is that if we have a solution \mathbf{x}_t of $H(\mathbf{x}, t) = \mathbf{0}$, then for a small perturbation h the solution \mathbf{x}_{t+h} of $H(\mathbf{x}_{t+h}, t+h) = \mathbf{0}$ will be close to \mathbf{x}_t , and hence it can be computed fast with Newton method using \mathbf{x}_t as initial values. Hence, as t varies smoothly from 1 to 0, \mathbf{x}_0 will smoothly converge to a root \mathbf{x}^* of F . Obviously, this process has to be iterated for each root \mathbf{x}_0 of G .

It remains to decide how to choose the starting system. Suppose F has n polynomials of degree d_1, \dots, d_n . By Bézout theorem [29], F can have at most $K = \prod_{i=1}^n d_i$ distinct roots. Hence choosing G to have exactly K distinct solutions will guarantee finding all solutions of F . A system satisfying this property can be easily constructed by choosing

$$G = \begin{bmatrix} x_1^{d_1} - a_1 \\ \vdots \\ x_n^{d_n} - a_n \end{bmatrix}, \quad (4)$$

where a_i are some non-zero complex numbers. The system $G(\mathbf{x}) = \mathbf{0}$ will then have exactly K solutions.

Now, for each root \mathbf{x}_0 of G , we apply homotopy continuation and track the path to the roots of F . It is good to notice that F needs not have K distinct solutions. Indeed, sparse systems are very likely to have strictly less distinct solutions than K . Practically, this means that a starting system built with (4) will introduce some computational overhead, as it will track more paths than necessary. Recently, more efficient initializations, exploiting polyhedral geometry, have been proposed for sparse systems [30], [31]. In our numerical experiments, we used the Julia implementation `HomotopyContinuation.jl` [32] and the polyhedral starting system proposed in [31] as initialization.

III. SOLVER DESIGN

Directly applying homotopy continuation to the equations as in (1) would result in a poorly conditioned solver. In this section we describe the algebraic manipulation that leads to a more stable formulation of the problem.

A. Parametrization of positions

In the general 2D case, node positions require $2m + 2n - 3$ unknowns. However, it was shown in [6], that for the case with $n = 3$, the positions can be parametrized using only five unknowns.

We start by defining the *compaction matrix* $\tilde{\mathbf{D}} \in \mathbb{R}^{(m-1) \times (n-1)}$ as

$$[\tilde{\mathbf{D}}]_{ij} = d_{i+1,j+1}^2 - d_{1,j+1}^2 - d_{i+1,1}^2 + d_{11}^2, \quad (5)$$

and fix the Gauge freedom by setting $\mathbf{r}_1 = \mathbf{0}$ and $\mathbf{r}_{2y} = \mathbf{0}$. The remaining points can be parametrized as follows [16]

$$\begin{aligned} \mathbf{s}_1 &= \mathbf{L}\mathbf{b}, \\ \mathbf{r}_i &= \mathbf{L}^{-\top} \tilde{\mathbf{D}}_{i-1}^{\top}, \quad i = 2 \dots m, \\ [\mathbf{s}_2 \ \mathbf{s}_3] &= \mathbf{L} \left(-\frac{1}{2} \mathbf{I} + [\mathbf{b} \ \mathbf{b}] \right), \end{aligned} \quad (6)$$

where \mathbf{b} is a vector of 2 unknowns and \mathbf{L} is a 2×2 upper triangular matrix, \mathbf{I} is the identity matrix and $\tilde{\mathbf{D}}_i$ denotes the i th row of $\tilde{\mathbf{D}}$. Defining the symmetric matrix $\mathbf{H} = (\mathbf{L}^{\top} \mathbf{L})^{-1}$ we can derive the equations

$$\begin{aligned} \text{(A)} \quad d_{11}^2 &= \mathbf{b}^{\top} \mathbf{H}^{-1} \mathbf{b}, \\ \text{(B)} \quad d_{1j}^2 - d_{11}^2 &= \frac{1}{4} \mathbf{e}_{j-1}^{\top} \mathbf{H}^{-1} \mathbf{e}_{j-1} - \mathbf{b}^{\top} \mathbf{H}^{-1} \mathbf{e}_{j-1}, \quad (7) \\ \text{(C)} \quad d_{i1}^2 - d_{11}^2 &= \tilde{\mathbf{D}}_{i-1} \mathbf{H} \tilde{\mathbf{D}}_{i-1}^{\top} - 2\mathbf{b}^{\top} \tilde{\mathbf{D}}_{i-1}^{\top}, \end{aligned}$$

where $[\mathbf{e}_1, \mathbf{e}_2]$ are the standard basis of \mathbb{R}^2 and $i = 2, \dots, m$ and $j = 2, 3$. Since \mathbf{H} is symmetric, it has only 3 unknowns. Recalling we also have 3 unknown offsets, the total number of unknowns will be 8. For a general configuration with m receivers and 3 transmitters we will have 1 equation of type (A), 2 equations of type (B) and $m - 1$ equations of type (C), hence $m + 2$ constraints in total.

B. Offset constraints

It can be shown [10], that the compaction matrix (5) must have rank 2. For configurations with $m \geq 4$ and $n \geq 4$, this means that some extra constraints can be imposed on the offsets by setting all 3×3 determinants in $\tilde{\mathbf{D}}$ to 0. While we can obtain $\binom{m-1}{3} \cdot \binom{n-1}{3}$ rank constraints in total, only $(m-3)(n-3)$ of those are independent. This approach was used in [6] to solve some overdetermined problems by first solving for the offsets separately and then localizing the nodes using synchronized network calibration techniques. Note that this approach alone is not suitable for smaller networks, but it can still offer extra constraints to add to (7).

During the experiments we will use both the initial formulation and the one described in this section. To distinguish between the two, we call the one using equations as in (1) the *primal formulation*, and the one using equations (7) and eventual offset constraints the *dual formulation*.

C. Summing up: building the homotopy solvers

The dual formulation significantly reduced the complexity of the problem and it can thus be efficiently solved by homotopy continuation. Here we discuss some solver-specific implementation details, to allow reproducibility of the experiments.

- **6r/3s:** This is a minimal configuration. Using the dual formulation, we obtain a total of 8 unknowns and $6 + 2 = 8$ equations like (7). This nonlinear polynomial system can now be solved with homotopy continuation. Next \mathbf{L} can be retrieved with Cholesky factorization and the positions can finally be computed with (6). Being a nonlinear minimal problem, the uniqueness is not guaranteed. However, not all the solutions of the dual formulation are necessarily solutions of the primal. Particularly, we accept only *real solutions* of the dual formulation, i.e. solutions which are real and lead to a positive definite matrix \mathbf{H} . However, some of the real solutions may still be false solutions. These can be pruned even more by substituting the real solutions into the primal formulation and accepting only those with a residual error lower than a given threshold.
- **7r/3s and 6r/4s:** For both instances, we first discard the extra receiver (for 7r/3s) or transmitter (for 6r/4s) and call the 6r/3s solver. For each candidate solution of the 6r/3s we trilaterate the extra receiver from 3 transmitters for 7r/3s, or trilaterate the extra transmitter and offset from 4 receivers for 6r/4s. Finally, we substitute the candidate solutions into the primal problem and output the one that leads to the smallest residual error.
- **5r/4s:** As mentioned in the introduction, this problem is interesting because it is not minimal, but it cannot be reduced to minimal configurations either. Using the dual formulation, this configuration has $5 + 4 = 9$ unknowns. From the whole compaction matrix, we obtain 2 independent rank constraints and leaving the last transmitter out, we can obtain 7 equations in \mathbf{H}, \mathbf{b} and the first 3 offsets, having 9 equations in total. We can thus solve the dual formulation using homotopy continuation and finally trilaterate the last transmitter from 3 receivers. As this problem is subminimal, the solution is unique. Hence, from the multiple solutions of the dual formulation, we accept the one that leads to the smallest residual error in the primal formulation.
- **5r/5s:** To solve this configuration, we simply leave out the last transmitter and offset, call the 5r/4s solver and finally trilaterate the last transmitter and offset from 4 receivers.

IV. RESULTS

In this section we discuss benchmarking of the solvers on synthetic data. The experiments are run with Matlab 2019b and Intel i7-8565U CPU@1.80GHz processor. Each solver ran in approximately 10 s.

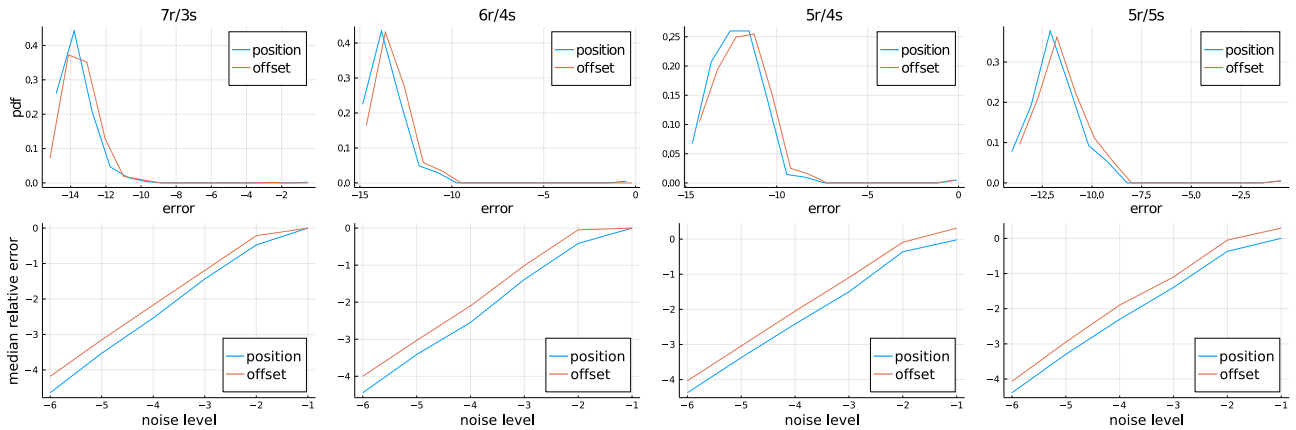


Fig. 1: Quantitative benchmark of our solvers. **Upper row:** relative error distributions for clean data. **Lower row:** median relative error at different noise levels, both in logarithmic scale.

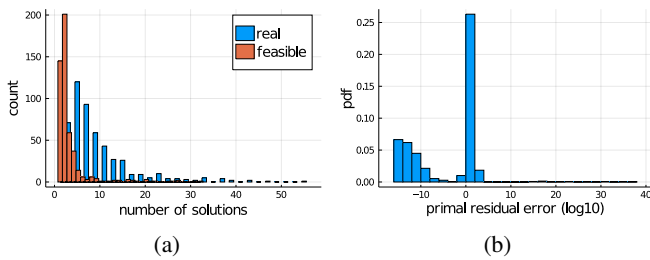


Fig. 2: Number of solutions and primal residual errors for 6r/3s solver.

A. Minimal solver: 6r3s

The 6r3s configuration is minimal and as such, it is guaranteed to have a finite, but not unique, number of solutions. It was speculated in [23] that the 6r3s configuration would have up to 150 solutions. To investigate this claim, we simulate this configuration with nodes at random positions and offsets and compute the number of real solutions obtained from the dual formulation. The distribution of the number of solutions is depicted in Figure 2a. As the histogram and Table II reveal, the dual formulation has from 2 to 55 real solutions. However, not all the real solutions of the dual are necessarily solutions of the primal. Indeed, Figure 2b shows the residual errors when the dual solutions are substituted into the primal system. As this histogram reveals, only some of the dual solutions are solutions of the primal problem and the others are false roots. Practically, this means that we can prune even more the set of real solutions by accepting only those that have a residual error in the primal problem less than a given threshold. In the experiments, we used a threshold of 10^{-10} and the distribution of the feasible real solutions is also reported in Figure 2a and Table II. Despite the number of solutions is significantly smaller than what was predicted in [23], still a unique solution cannot be obtained without some extra knowledge of the system. Furthermore, removing the false solutions thresholding the residuals in the dual formulation works fine for clean data.

TABLE II: Number of solutions of the 6r/3s solver

	min	max	mean	st. deviation
real solutions	2	55	8.7	7.6
feas. solutions	1	32	3	3.9

In the presence of noise false roots may also be included, or feasible roots excluded. For this reason, in the next section we also examine subminimal configurations, for which the uniqueness of the solution is guaranteed.

B. Subminimal solvers

We first benchmark our solvers with clean data, by generating hundreds of random instances of the problems, using a Gaussian distribution for both position and offset. The distributions for position and offset errors are depicted in Figure 1. As can be noticed, the solvers can achieve very accurate solutions, with the relative error in the order of magnitude 10^{-12} . We also study how our solvers perform with noisy measurements. As the lower row of Figure 1 shows, our homotopy solver is alone stable also for noisy data even without nonlinear refinement afterwards.

V. CONCLUSIONS

This paper addressed the problem of sensor networks self-calibration with unsynchronized transmitters. Due to the high number of unknowns and degree of the system, this is a computationally challenging problem to solve. We showed that homotopy continuation offers a powerful tool to overcome these challenges and developed new solvers, stable both for clean and noisy data, which allowed to solve previously unsolved configurations. This opens several perspectives for the applications of homotopy continuation in the domain of network calibration and localization algorithms in general.

ACKNOWLEDGMENTS

This work was partially funded by the Academy of Finland project 327912 REPEAT and the Swedish strategic research project ELLIIT.

REFERENCES

- [1] E. Miluzzo, N. D. Lane, A. T. Campbell, and R. Olfati-Saber, "Calibree: A self-calibration system for mobile sensor networks," in *Distributed Computing in Sensor Systems*. Springer Berlin Heidelberg, 2008.
- [2] J. Wendeberg and C. Schindelbauer, "Polynomial time approximation algorithms for localization based on unknown signals," in *Algorithms for Sensor Systems*, A. Bar-Noy and M. M. Halldórsson, Eds. Springer Berlin Heidelberg, 2013.
- [3] H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh, "Collaborative beamforming for distributed wireless ad hoc sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 11, pp. 4110–4124, 2005.
- [4] S. Thrun, "Affine structure from sound," *Advances in Neural Information Processing Systems*, vol. 18, pp. 1353–1360, 2005.
- [5] M. Crocco, A. Del Bue, and V. Murino, "A bilinear approach to the position self-calibration of multiple sensors," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 660–673, 2011.
- [6] Y. Kuang, S. Burgess, A. Torstensson, and K. Åström, "A complete characterization and solution to the microphone position self-calibration problem," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.
- [7] V. C. Raykar, I. V. Kozintsev, and R. Lienhart, "Position calibration of microphones and loudspeakers in distributed computing platforms," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 1, pp. 70–83, 2005.
- [8] S. Burgess, Y. Kuang, and K. Åström, "Node localization in unsynchronized time of arrival sensor networks," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 2012, pp. 2042–2046.
- [9] R. L. Moses, D. Krishnamurthy, and R. M. Patterson, "A self-localization method for wireless sensor networks," *EURASIP Journal on Advances in Signal Processing*, vol. 2003, no. 4, pp. 1–11, 2003.
- [10] Y. Kuang and K. Åström, "Stratified sensor network self-calibration from tdoa measurements," in *21st European Signal Processing Conference (EUSIPCO 2013)*, 2013.
- [11] M. Pollefeys and D. Nister, "Direct computation of sound and microphone locations from time-difference-of-arrival data," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 2445–2448.
- [12] R. Biswas and S. Thrun, "A passive approach to sensor network localization," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 2. IEEE, 2004, pp. 1544–1549.
- [13] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-free distributed localization in sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 340–341.
- [14] M. Larsson, G. Flood, M. Oskarsson, and K. Åström, "Upgrade methods for stratified sensor network self-calibration," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 4851–4855.
- [15] K. Batstone, G. Flood, T. Beleyur, V. Larsson, H. R. Goerlitz, M. Oskarsson, and K. Åström, "Robust self-calibration of constant offset time-difference-of-arrival," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 4410–4414.
- [16] L. Ferranti, K. Åström, M. Oskarsson, J. Boutellier, and J. Kannala, "Sensor networks tdoa self-calibration: 2d complexity analysis and solutions," 2020.
- [17] Z. Kukulova, M. Bujnak, and T. Pajdla, "Automatic generator of minimal problem solvers," in *European Conference on Computer Vision*. Springer, 2008, pp. 302–315.
- [18] V. Larsson, K. Åström, and M. Oskarsson, "Efficient solvers for minimal problems by syzygy-based reduction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 820–829.
- [19] Z. Kukulova, M. Bujnak, and T. Pajdla, "Polynomial eigenvalue solutions to minimal problems in computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1381–1393, 2011.
- [20] A. Morgan and A. Sommese, "Computing all solutions to polynomial systems using homotopy continuation," *Applied Mathematics and Computation*, vol. 24, no. 2, pp. 115–138, 1987.
- [21] R. Fabbri, T. Duff, H. Fan, M. H. Regan, D. d. C. d. Pinho, E. Tsigaridas, C. W. Wampler, J. D. Hauenstein, P. J. Giblin, B. Kimia *et al.*, "Trppl-trifocal relative pose from lines at points," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 073–12 083.
- [22] T. Duff, K. Kohn, A. Leykin, and T. Pajdla, "P11p-point-line minimal problems under partial visibility in three views," *arXiv preprint arXiv:2003.05015*, 2020.
- [23] H. Stewenius, "Gröbner basis methods for minimal problems in computer vision," 2005.
- [24] J. Kileel, "Minimal problems for the calibrated trifocal variety," *SIAM Journal on Applied Algebra and Geometry*, vol. 1, no. 1, pp. 575–598, 2017.
- [25] D. M. Malioutov, M. Cetin, and A. S. Willsky, "Homotopy continuation for sparse signal representation," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 5, 2005, pp. v/733–v/736 Vol. 5.
- [26] J. Fanjul, González, I. Santamaria, and C. Beltrán, "Homotopy continuation for spatial interference alignment in arbitrary mimo x networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 7, pp. 1752–1764, 2017.
- [27] T. Wayburn and J. Seader, "Homotopy continuation methods for computer-aided process design," *Computers Chemical Engineering*, vol. 11, no. 1, pp. 7–25, 1987.
- [28] Y. Chang and J. Seader, "Simulation of continuous reactive distillation by a homotopy-continuation method," *Computers Chemical Engineering*, vol. 12, no. 12, pp. 1243–1255, 1988.
- [29] D. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry*. Springer Verlag, 1998.
- [30] B. Huber and B. Sturmfels, "A polyhedral method for solving sparse polynomial systems," *Mathematics of computation*, vol. 64, no. 212, pp. 1541–1555, 1995.
- [31] A. N. Jensen, "Tropical homotopy continuation," *arXiv preprint arXiv:1601.02818*, 2016.
- [32] P. Breiding and S. Timme, "Homotopycontinuation.jl: A package for homotopy continuation in julia," in *International Congress on Mathematical Software*. Springer, 2018, pp. 458–465.